recombinant DNA or genetic engineering. His accomplishment generated widespread intellectual excitement among scientists because of its revolutionary nature and the seeming potential of DNA to outperform conventional computers. Some scientists saw its power in its ability to encode information in such a small space, and by manipulating the strands, to sort through an exhaustive library of all possible answers to problems of a certain size.

The Gene-Based Computer

DNA is a major constituent of the chromosomes that determine our heredity. It consists of two long molecular chains twisted into the form of a double helix, and joined by hydrogen bonds (figure 5.1). The bonds are linked to four nucleotides, adenine and thymine (A and T) or cytosine and guanine (C or G). In living organisms, the bases in each of these nucleotides bonds with its complement—A to T and C to G—in a pattern that ultimately determines our heredity. (A base is a chemical compound, distinguished from an acid, that contains oxygen and hydrogen bound together to form a hydroxyl ion, and has a pH greater than 7; a common base is sodium hydroxide, NaOH.) Two DNA strings are complementary if the second has a sequence such that A and T are interchanged, and C and G are interchanged.

In order to compute using strands of DNA, one must first start by elucidating, step by step, the computer algorithm that solves the problem of interest. Then, one must translate each of these steps into the language of DNA, by determining the sequences and exact manipulations to be performed on the DNA. The DNA starting material with the proper sequences must be custom made in a lab; these short strands are known as oligonucleotides. This process usually involves the close collaboration of computer scientists and biologists; the cross-pollination that typically results has helped foster growth in the field.

The DNA-based computation can be arranged as a series of test tubes filled with water and up to 10^{20} strands of DNA. The strands consist of the preselected sequence of oligonucleotides, which are then manipulated and sorted according to the algorithm. For example, they can be sorted by length, or by performing various "and," "or," or "not" operations to check for the presence of certain sequences in an individual strand of

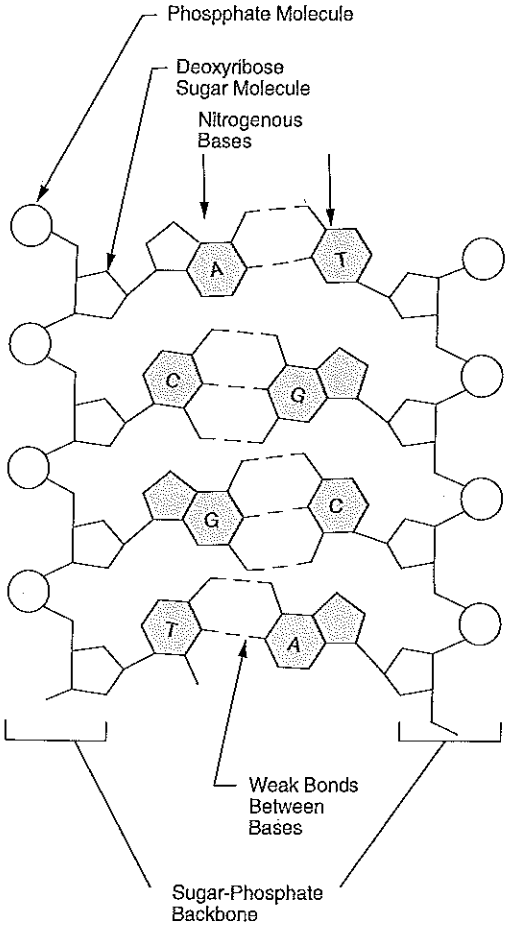

Figure 5.1
DNA structure. The four base pairs of DNA are arranged along the sugar-phosphate backbone in a particular order, encoding all genetic instructions for an organism. The bases adenine (A) pairs with thymine (T), while cytosine (C) pairs with guanine (G). The two DNA strands are held together by weak bonds between the bases. (Source: US Department of Energy Human Genome Project, <http://www.ornl.gov/hgmis>)

DNA. The strands remaining at the end of these operations are amplified using standard molecular biology lab techniques, and then are sequenced to identify the resulting DNA sequence that represents the answer to the problem (figure 5.2). In principle, only one strand containing the answer needs to be amplified.

Each operation performed sequentially on the DNA creates a new batch from the results of earlier mixtures; the operations can consist of separating the strands by length, pouring the contents of one test tube into another, extracting those strands with a given sequence, heating or cooling them, or using enzymes to splice the DNA. The series of test tube manipulations forms what computer scientists call "a single-instruction, multiple-data" (SIMD) computation performed in parallel on the DNA.

The numerous laboratory techniques of molecular biology, known as recombinant DNA techniques, enable a broad range of algorithmic operations to be used in DNA computing. These methods have been developed over the last few decades for use in biology wet labs for genetic engineering purposes. The principle laboratory operations used in DNA computing are the following:

- Synthesizing DNA strands of a desired sequence of A, T, G, or C and length.
- Separating double-stranded DNA into single strands by heating, breaking the hydrogen bonds that connect them. This is called *denaturation.*
- Fusing single-stranded DNA into double-stranded DNA with complementary base pairs connected—called *annealing* or *hybridization.*
- Removing all the single strands or all the double strands from the mixture.
- Cutting the strands in exact places using restriction enzymes. (An enzyme is a catalyst that causes a chemical reaction to occur.) Many bacteria produce restriction enzymes, protecting the cell by cleaving and destroying the DNA of invading viruses. Each restriction enzyme cleaves the DNA at a particular site, so each operation requires a different one.
- Separating the strands by length.
- Extracting and separating out strands with a known sequence of 15–20 base pairs anywhere along the strand.
- Finding a particular string encoded in a DNA strand.
- Constructing complementary strands for short (15–20 base pairs) strings in a strand.

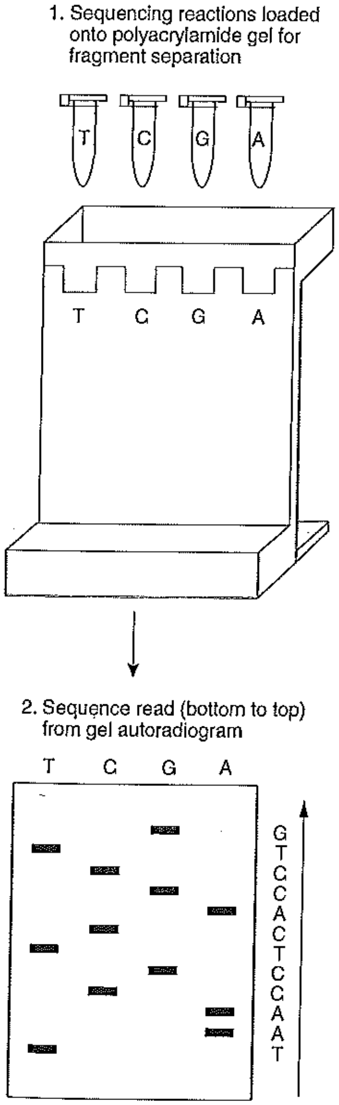
1. Sequencing reactions loaded onto polyacrylamide gel for fragment separation
T
C
G
A
T C G A
2. Sequence read (bottom to top) from gel autoradiogram
T C G A
G
T
C
C
A
C
T
C
C
A
A
T

Figure 5.2
DNA sequencing. This particular method of sequencing (called the Sanger method) uses enzymes to synthesize DNA chains of varying lengths, stopping DNA replication at one of the four bases and then determining the resulting fragment lengths. The fragments are then separated by a method called electrophoresis (1) and the positions of the nucleotides analyzed to determine sequence (2). (Source: US Department of Energy Human Genome Project, <http://www.ornl.gov/hgmis>)

- Making many copies of a given strand of DNA using the polymerase chain reaction (PCR). To use PCR, double-stranded DNA is heated and separated, and then short strands that are complementary to sequences in the targeted DNA strand (primers) are added, together with free base pairs and the enzyme DNA polymerase, which catalyzes the elongation of a piece of DNA. In a series of heating and cooling cycles, the DNA doubles with each cycle.
- Appending a given string of DNA to a selected substring, or to the entire strand of DNA present, called *ligation.*
- Destroying a marked strand of DNA.
- Detecting and reading. This enables one to determine if a test tube of DNA does/doesn't contain at least one strand with the desired result; and, if it does, to interpret its sequence.

The Traveling Salesman

Adleman's first successful test of the theory of DNA computing was solving a well-known problem in computing called the Directed Hamiltonian Path Problem (or "Traveling Salesman Problem"). According to Adleman's method, a particular DNA sequence can represent the vertices or endpoints of a graph, and one can describe random paths through the graph by performing operations on the various strands of DNA. In this case, Adleman chose seven cities to represent the vertices of the graph, and fourteen links connecting the cities in various ways that represent one-way non-stop flights between two cities. The problem entails figuring out whether there is a route that takes the salesman from a given starting city to a given end city, passing through each city once and only once (figure 5.3).

Step one is to assign each city a unique six-letter name, where each letter stands for a given nucleotide (adenine, guanine, cytosine, and thymine, or A, G, C, and T). So in this case we do the following:

San Francisco = AGCTAT
New York = GTTAGC
Seattle = ACACTA
Sydney = AGTTAT
Seville = TATCTC
London = TAAAAG
Rome = GATGCT

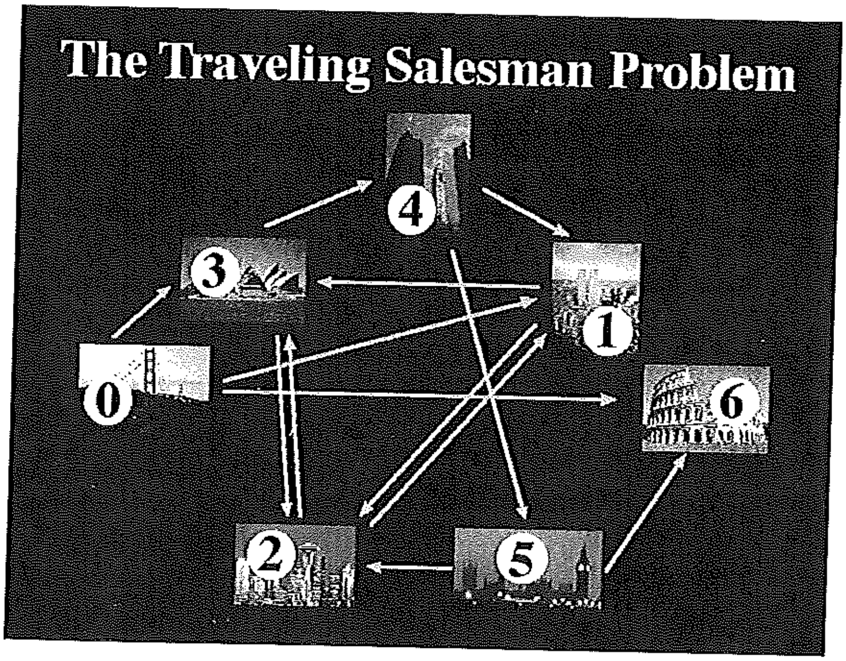

Figure 5.3
DNA computing solved the seven-city directed-path problem by reacting coded DNA strands, as represented by the letter sequences next to the pictures, to imprint a successful route on the product material. (Source: L. Kari and L.F. Landweber, "Computing with DNA," in *Bioinformatics Methods and Protocols*, S. Misener and S. Krawetz, eds., Humana, Totowa, NJ, 1999, p. 415)

Then one assigns the corresponding flight names for those cities which have direct connections, using the first three letters of the city name to signify arrival and the last to indicate departure, for example:

Departs San Francisco, arrives Sydney = TATAGT
Departs Sydney, arrives London = TATTAA

and so on. The four nucleotides only bond chemically with their complementary base pair, so each city code also possesses a complementary name, for example:

San Francisco (original code) = AGCTAT
San Francisco (complement code) = TCGATA
Sydney (original code) = AGTTAT
Sydney (complement code) = TCAATA

Single strands of DNA containing all the direct flight names and other strands containing all the complementary city names are then synthesized in a lab and mixed together in a test tube, so if a strand containing a flight from San Francisco to Sydney encounters another with the Sydney complement city name, the two will bond:

```
TATAGT
   |||
   TCAATA
```

Then, if this strand bonds in the test tube with one containing a flight from Sydney to, say, Seville, we have:

```
TATAGTTATTAT
   ||||||
   TCAATA
```

and when this piece runs into another containing the Seville complement name, we get:

```
TATAGTTATTAT
   |||||||||
   TCAATAATAGAG
```

which will react with a strand for a direct connection between Seville and another city, and so on. The next step is to read out the strands containing the answer, which results from first filtering, measuring, and reading out the DNA in the test tubes so that only strands beginning with San Francisco and ending with Rome, and that contain all the five other city codes and are seven city codes long, remain in the test tube. The sequence in these strands will contain the answer.

The remarkable parallelism of the DNA computer uses brute force to try out all the possible solutions to the problem. It does so in about 10^{14} operations per second (assuming the binding of two DNA molecules is a single operation, and also that approximately 10^{14} copies of the associated oligonucleotide are added to represent each direct flight, and that half of these bind).

The traveling salesman problem forms part of a class of problems called "NP-complete," where "NP" stands for non-deterministic polynomial time. This means a set of problems whose answers can be checked in polynomial time, that is, where the computer running time needed to verify if an answer is correct or not is bounded by a polynomial function. (This is assuming that a genie has already revealed the correct answer to be checked!)

NP-complete problems, a special subset of NP problems, currently have no known polynomial time solution. What makes Adleman's accomplishment so remarkable is that by successfully demonstrating that the traveling salesman problem could be solved with DNA, he showed, as proof of concept, that the biological computer was capable of solving a small-scale NP-complete problem. (This does not mean, however, that all NP problems can be feasibly solved.)

Adleman's experiment set a completely new direction in bio-inspired computation. Even though his approach could not be scaled up indefinitely to very large combinatorial search problems as the number of individual DNA strands needed tends to grow exponentially with the size of the problem, it was a successful demonstration of a paradigm shift in computing. Moreover, it showed that individual molecules could be manipulated in the lab to implement complex algorithms—a result that those in the emerging field of nanoscience found very encouraging.

The Fledgling Field Advances

Adleman's experiment helped attract many talented scientists to the field from biology, computer science, physics, chemistry, and engineering, and growth continues in both theory and experiments with DNA computing. One type of problem that several researchers have found very amenable to DNA computations is the satisfiability problem (SAT), an important NP-complete problem in computer theory.

Solving SAT problems with DNA demonstrates how DNA computers work as powerful search engines. Satisfiability problems attempt to answer, for a given Boolean expression A, if values can be assigned to the variables in A that would make A a true statement. With DNA, any strand that encodes values that make the expression true is a potential solution. For an SAT problem with n variables, an electronic computer must test 2^n variables one by one, which can require a lot of time for large n. One theoretical approach to SAT with DNA proposed by Georgia Tech computer scientist Dick Lipton, however, can check all the variables simultaneously, providing a much more efficient method for solving the problem.